FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

Untersuchung des TIS Firewall Toolkit als Application Gateway im Forschungszentrum Jülich

Ralph Niederberger

KFA-ZAM-IB-9617

Juli 1996
(Stand 24.07.1996)

Inhaltsverzeichnis

Verzeichnis der Abbildungen . . . . . **v**

1 Einführung **1**

2 Motivation **2**

3 Das TIS Firewall Toolkit **3**

4 TIS Firewall Toolkit — Softwarekomponenten **4**
4.1 Smap: SMTP Service 4
4.2 NETACL 4
4.3 FTP-GW: Proxy Server für FTP 5
4.4 TN-GW: Proxy Server für Telnet 5
4.5 HTTP-GW: Proxy Server für HTTP 5
4.6 X-GW: Proxy Server für X-Applikationen 5
4.7 login-sh, rlogin-gw und plug-gw 6
4.8 Syslogd: System Logging 6
4.9 Authd: Netzwerk-Authentication Service 6

5 Hardwarekomponenten und Netzlayout **7**

6 Installation, Konfiguration und Nutzbarkeit der einzelnen Dienste **9**
6.1 E-Mail 10
6.2 FTP 11
6.3 Telnet 13
6.4 WWW 15
6.5 X-Protocol 16

7 Zusammenfassung der Untersuchung **18**

8 Anhang **20**
8.1 Modifizierte und zusätzlich notwendige Dateien 20
8.2 TIS Internet Firewall Toolkit License Agreement 24
8.3 TIS Internet Firewall Toolkit — Disclaimer 26
8.4 Stichwortverzeichnis 27
8.5 Literatur 30

Verzeichnis der Abbildungen

Abb. 1 Konzept eines Screened Subnet mit demilitarisierter Zone (DMZ) . . . 3
Abb. 2 Physikalischer Aufbau des Firewall-Systems 7
Abb. 3 Schematischer Netzaufbau 8
Abb. 4 mailrelay.zam.kfa-juelich.de 10
Abb. 5 ftp-gw.zam.kfa-juelich.de 12
Abb. 6 tn-gw.zam.kfa-juelich.de 14
Abb. 7 WWW-(W3)-Service 15
Abb. 8 X Window System Architektur 16

1 Einführung

TCP/IP stellt heute das am meisten benutzte Netzwerk Protokoll im wissenschaftlich technischen Bereich in der KFA dar.

Derzeit sind auf dem Gelände der KFA ca. 4000 Rechner, vom Entry-Level-PC bis zu den Super-Computern CRAY T90, CRAY M/94 und Intel Paragon, installiert und an KFAnet/Internet angeschlossen.

Haupttransportmedium zu den KFA Instituten mit hohen Brandbreiteanforderungen ist derzeit FDDI mit 100 Mbits/s und Ethernet mit 10 Mbit/s andernfalls. ATM ist in der Planung, Vorbereitung und Testphase.

Die derzeit zur Verfügung gestellten Netzwerkdienste sind FTP, Telnet, Elektronic Mail, Remote Printing, NFS, NetNews und WWW. Weitere Applikationen, basierend auf dem BSD Socket Interface, sind in Benutzung.

Für Backup und Archivierung wird ADSM (ADSTAR Distributed Storage Manager) benutzt.

Da den System-Administratoren der am Netz installierten Arbeitsplatzrechner keine allgemeingültigen Vorschriften bezüglich der Sicherheitskonfiguration gemacht werden können, ist eine Sicherheits-Politik auf Rechner-Ebene nicht durchsetzbar. Eine Kontrolle der angeschlossenen Systeme kann aufgrund der Zahl der installierten Rechner nicht durchführt werden. Ein Ansatz auf Rechner-Basis kann somit nur zusätzliche Sicherheit erbringen. Eine zentrale Sicherheitsschranke, die für alle Systeme gleichzeitig wirkt, wäre daher notwendig.

Heutige Netzwerkattacken sind wesentlich intelligenter und technisch komplexer angelegt, als dies noch vor 5–6 Jahren der Fall war. Um diese Attacken abzuwenden, muß jedes verfügbare Mittel genutzt werden. Eines dieser Mittel ist ein Firewall.

Firewalls bieten derzeit die beste Methode, ein Netz und die daran angeschlossenen Rechner vor Angriffen von aussen zu sichern. Typischerweise wird ein Firewall an der Grenze zwischen internem Netz und externen Rechnern installiert. Ein Firewall beschränkt den Verkehr zwischen externen und internen Rechnern auf die vordefinierten erlaubten Aktionen. Die Existenz eines Firewalls reduziert in großem Maße die Gefahren, die externe Cracker gegenüber lokalen Rechnern darstellen [1].

Ein Firewall setzt sich aus verschiedenen Komponenten zusammen. Eine oder mehrere Packet-Screens überprüfen den Datenverkehr durch Access-Listen auf erlaubte Transaktionen. Alle Transaktionen, die nicht durch Access-Listen ausreichend geschützt werden können, werden über Applikation-Gateways angeboten [1,5].

Der vorliegende Bericht untersucht das sogenannte Firewall Toolkit der Firma Trusted Information Systems (TIS-FWTK), eine Sammlung von Programmen benötigter Dienste, die das Applikation Gateway zur Verfügung stellen sollte [2,3,4].

2 Motivation

Die Untersuchung zu “Firewalls – Sicherheit und Benutzerakzeptanz in Forschungsnetzen” [5] hat ergeben, daß die Applikationen Telnet, FTP, HTTP und X ausreichend nur durch Applikation-Gateways gesichert werden können.

Allgemein kann man das eigene Netz vor Übergriffen von außen auf drei Arten schützen:

Roll your own: Man nehme selbst geschriebene Überwachungsprogramme, ergänze diese um Hilfs-, Logging- und Accountingprogramme sowie öffentlich frei verfügbare Zusatzprogramme und füge diese zu einem kompletten Firewall-Gesamtsystem zusammen. Stehen sowohl Talent, Know-How als auch Zeit und Ressourcen zur Verfügung, so stellt diese eine gute Lösung des Sicherheitsproblems dar.

Black Box: Man kaufe eine Gesamtlösung von einem Firewall-Hersteller oder Anbieter. Die vorgefertigten Programme werden von der unterstützenden Firma auf die lokalen Gegebenheiten angepaßt. Die Programmpakete sind meist nicht durch eigene Entwicklungen ergänzbar, stellen somit eine eigene für sich alleinstehende Lösung dar. Soll die Sicherheit des Netzes in die Hände einer externen Firma gelegt werden, lokale Arbeit vermieden werden, und stehen genügend Finanzmittel zur Verfügung für Kauf, Installation und Wartung, so ist dies eine gangbare Alternative.

Crystal Box: Diese ist charakterisiert durch Quellcode der Komponenten, Sicherheitsanalyse, Dokumentation, Erweiterbarkeit durch lokale Prozeduren. Eine *Crystal-Box* erlaubt es, Software– und Hardwarekomponenten unabhängig voneinander zu testen, zu erweitern und auf lokale Gegebenheiten anzupassen. Dies kann durch eigene Mitarbeiter oder durch Fremdfirmen geschehen. Aus Sicht des Sicherheitsexperten ist diese Form eines Firewalls aufgrund der sie charakterisierenden Eigenschaften die Beste der drei möglichen Alternativen.

Den drei möglichen Methoden der Firewall-Implementierung stehen zwei Design-Philosophien gegenüber.

Alles, was nicht ausdrücklich verboten ist, ist erlaubt. Diese Philosophie, die den Benutzern am angenehmsten ist, erfordert größte Aufmerksamkeit des Firewall-Managers. Er muß frühzeitig, d.h. bevor die Angreifer das Ziel erreicht haben, die Sicherheitslücken schließen. Neue Sicherheitsprobleme müssen sofort analysiert und gegebenenfalls durch Eingriffe in die Sicherheitsstrategie gelöst werden. Dies geschieht normalerweise durch Access-Listen in Routern.. Bei diesem klassischen Ansatz werden erst Gegenmaßnahmen ergriffen, wenn eine Sicherheitslücke erkannt wird, .

Alles, was nicht ausdrücklich erlaubt ist, ist verboten. Hier wird dem externen Angreifer ein nur minimales Angriffsziel geboten. Alle nicht benötigten Dienste sind abgeschaltet oder durch Access-Listen geschützt; sie können daher nicht für Attacken mißbraucht werden. Dieser konservative, eventuell auch paranoide, aber sichere Ansatz erfordert den Eingriff des Firewall-Managers immer dann, wenn weitere Services benötigt werden.

Das TIS-Firewall Toolkit stellt eine kostengünstige und als Crystal Box konzipierte, allgemein akzeptierte Lösung für Applikation Gateways dar, das zusammen mit entsprechend konfigurierten Access-Listen nach dem Prinzip “***Alles, was nicht ausdrücklich erlaubt ist, ist verboten”*** arbeitet. Daher wird im Folgenden eine Untersuchung dieses Produktes auf Funktion, Konfiguration und Benutzbarkeit durchgeführt.

3 Das TIS Firewall Toolkit

Das ***TIS Firewall Toolkit*** ist ein lizensiertes, aber kostenlos erhältliches Paket (nicht Public Domain), bestehend aus einer Ansammlung von Programmen und Design-Hilfen für den problemlosen Entwurf eines Netzwerk Firewalls. Die einzelnen Komponenten können gemeinsam, aber auch getrennt voneinander benutzt werden und sind auf den gängigen Unix-Systemen mit TCP/IP lauffähig.

Die Installation des Toolkits setzt praktische Erfahrung in der Unix System Administration und TCP/IP Netzwerk Konfiguration voraus. Das Toolkit ist keine schlüsselfertige Lösung, sondern erfordert die Anpassung an die jeweilige Netzwerk- und System-Umgebung. Durch unterschiedliche Konfigurationsentscheidungen können verschiedenartige Sicherheits-Level erreicht werden. Aufgabe des Firewall Administrators ist es, die für die lokale Umgebung bestgeeignete Sicherheits-Policy zu definieren und mit Hilfe des Toolkits zu konfigurieren. Das Toolkit selbst kann aber zur Entwicklung der Sicherheits-Policy keinen Beitrag leisten.

Die allgemeine Design-Philosophie des Toolkits enthält die folgenden Paradigmen:

- Jeder eventuell noch vorhandene Fehler darf die Funktionsfähigkeit des Toolkits (der Sicherheitspolicy) nicht beeinflußen.
- Privilegierte Dienste dürfen nicht von Rechnern außerhalb des geschützten Netzwerkbereiches erreichbar sein.
- Netzwerk-Dienste müssen grundsätzlich mit minimalen Privilegien laufen.
- Die Korrektheit des Systems muß überprüfbar sein.

Vor Installation des Toolkit muß festgelegt werden, was zu schützen ist, wie dies zu geschehen hat und wie dies im Einklang mit bestehenden Organisationzielen zu erreichen ist.

Der Ansatz des TIS Toolkits geschieht, wie oben erwähnt, allgemein gemäß dem Vorsatz:

Was nicht explizit erlaubt ist, ist verboten.

Als konzeptionelle Grundlage eines Firewalls im Forschungszentrum Jülich wurde in [5] die Konstruktion mittels eines ***Screened Subnet Gateways*** vorgeschlagen und somit für die TIS-Toolkit-Tests gewählt.

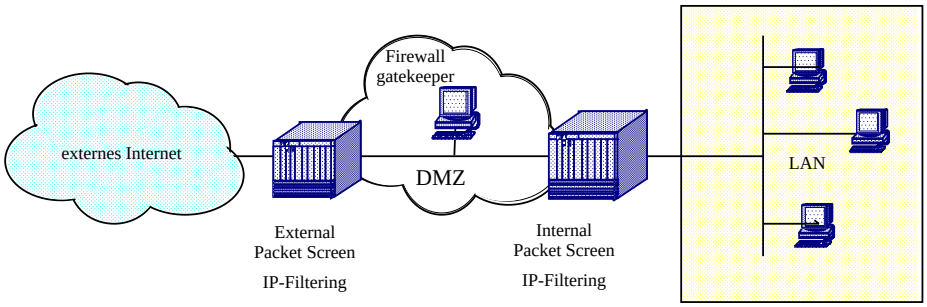

Abb. 1: Konzept eines Screened Subnet mit demilitarisierter Zone (DMZ)

Hierbei liegt das Firewall in einer *Demilitarisierten Zone* (*DMZ*, auf einem eigenen Subnetz) zwischen zwei Routern, die Pakete mittels Access-Listen überprüfen [5]. Hierdurch können unkritische Applikationen am Firewall vorbei geleitet werden, während kritische Applikationen über sogenannte ***Applikation Gateway Services*** realisiert werden. Authentisierung erlaubt es, diese Dienste zusätzlich durch Zugangsberechtigungen zu sichern.

4 TIS Firewall Toolkit — Softwarekomponenten

Das Firewall Toolkit stellt mehrere Software-Programme als Application-Gateways zur Verfügung. Alternativ können hier andere Public Domain erhältliche oder selbst geschriebene Programme benutzt werden.

4.1 Smap: SMTP Service

In den derzeitigen Unix-Systemen läuft der SMTP-Service (Build-In Programm zu *sendmail*) mit weitgehender Rootberechtigung ab. Der *sendmail*-Prozess wurde daher und aufgrund seiner großen Komplexität oft als Angriffsziel für Attacken benutzt [12].

Im Firewall Toolkit werden dem Sendmail-Prozess zwei Programme ***smap*** und ***smapd*** vorgeschaltet, die mit minimalen Privilegien in einem restriktiven Bereich als unprivilegierte Benutzer laufen (*chroot*[1]). Die beiden Prozesse kümmern sich nur um ankommende elektronische Nachrichten. Diese werden an den *sendmail*-Prozess weitergeleitet, der das Groß der Arbeit auch weiterhin durchführt. Mail-Versendung von lokalen Rechnern zu außerhalb gelegenen Rechnern wird auch weiterhin von diesen selbst und nicht vom Applikation-Gateway übernommen.

Der *smap*-Prozeß nimmt amkommende Mails an und speichert diese in ein reines Datenverzeichnis. Weiterhin protokolliert er den Mailempfang. Für beide Aufgaben braucht er keine Sonderrechte, sodaß der *smap*-Prozeß ohne Privilegien auskommt. Diese Tatsache und die feste Vorgabe des Ablageverzeichnisses verhindern, daß ein Angreifer per Mail Programme des Applikation-Gateways verändern oder gar ersetzen kann.

Das smap-Programm ist aufgrund seiner einfachen, klar definierten Aufgabe viel kürzer als das *sendmail*-Programm (ca. 700 statt 20.000 Codezeilen) und dadurch viel einfacher auf Sicherheitsrisiken zu überprüfen.

4.2 NETACL

Da der Prozess *inetd* in Unix-Systemen keine Möglichkeit der Zugriffskontrolle bietet, stellt das Firewall-Toolkit mit ***Netacl*** eine solche zur Verfügung. ***Netacl*** bietet diese Zugriffskontrolle für jeden Dienst einzeln basierend auf Internet-Adressen. So kann z.B. ein Benutzer von Rechner *a* einen anderen Telnet-Prozeß angeboten bekommen als ein Benutzer von Rechner *b*. Ebenso können die Authentifizierungsroutinen für die beiden Rechner unterschiedlich sein. Eine größtmögliche Flexibilität in der Sicherheitspolitik wird somit zur Verfügung gestellt.

So wird es in einer normalen Firewall-Umgebung nur wenigen Rechnern erlaubt sein, direkt auf dem Firewall ein Login duchzuführen, während alle anderen mit mehr oder weniger aufwendiger Authentifizierung nur das Telnet-Gateway ***TN-GW*** (siehe unten) benutzen können.

---

[1] Nach Aufruf von *chroot* kann ein Prozeß nicht auf Daten zugreifen, die oberhalb seiner Directory-Struktur liegen [6]. Sei der Prozess x chrooted nach /tmp/spec_dir/root. Ein Zugriff auf die Datei /etc/passwd durch den Prozess x bewirkt den tatsächlichen Zugriff auf /tmp/spec_dir/root/etc/passwd. Systemdateien können somit nicht durch den Prozess x verändert werden. Dies führt zu einer wesentlichen Erhöhung der Systemsicherheit.

Da die Sicherheit des Programms ***Netacl*** auf IP-Adressen basiert, kann es vor IP-Spoofing[2] nicht schützen. Diese Attacken müssen auf der vorgelagerten Packet Screen abgefangen werden [5].

4.3 FTP-GW: Proxy Server für FTP

Ein FTP-Gateway-Server im Firewall-System bietet die Möglichkeit, Dateien von extern nach intern und umgekehrt über das Firewall zu transferieren, ohne die Sicherheit des Firewalls zu beeinflussen. Hierzu wird wiederum ein nichtprivilegierter Prozess *ftp-gw* gestartet (*chroot*).

Der Proxy-FTP-Server bietet Zugriffskontrolle basierend auf IP-Adressen und/oder Usernamen (vergleiche *Netacl*) und kann einzelne FTP-Unterkommandos blockieren. Sicherheitsprobleme auf dem Firewall-Rechner können nicht auftreten, da der Prozeß nur die Konfigurationsdateien liest und alle Aktionen in einer speziellen Log-Datei protokolliert, ansonsten aber keine weiteren Dateizugriffe (I/O) macht.

4.4 TN-GW: Proxy Server für Telnet

Alle für das FTP-GW gemachten Aussagen treffen auch auf das Telnet-Gateway TN-GW zu. Zugriffskontrolle, basierend auf IP-Adressen und/oder Usernamen (vergleiche *Netacl*), helfen auch hier, Risiken bei der Kommunikation mit externen Rechnern auszuschalten.

4.5 HTTP-GW: Proxy Server für HTTP

Der HTTP-GW-Proxy-Server bietet internen Benutzern die Möglichkeit, auf externe HTTP-Server zuzugreifen. Umgekehrt können aber auch externe WWW-Nutzer auf WWW-Server zugreifen, die eventuell innerhalb des durch den Firewall geschützten Bereich liegen. Während der Zugriff von außen nach innen nur beschränkt erlaubt sein sollte, da ein WWW-Server immer ein potentielles Risiko darstellt und somit besser außerhalb des lokalen Netzes installiert werden sollte, kann der Zugriff von internen Rechnern auf externe Server einfach und transparent durch das HTTP-GW Gateway realisiert werden. Zugriffe geschehen, indem Anfragen über das Firewall HTTP-Gateway weitergeleitet werden. Auch hier können wieder die berechtigten Rechner (Internet-Adressen) eingetragen werden.

4.6 X-GW: Proxy Server für X-Applikationen

Das Programm X-GW bietet eine User-Level-X-Verbindung unter der Kontrolle des TN-GW-Dienstes. Dies geschieht, indem zuerst eine Telnet-Verbindung mit dem Firewall-Server aufgebaut wird. Durch Eingabe des Kommandos ***x-gw*** *hostname* wird dem Telnet-Gateway-Server mitgeteilt, daß eine X-Verbindung gewünscht wird. Der Server teilt daraufhin den Port mit, über den die X-Verbindung laufen kann (*display port=firewall:portnumber*). Der Client setzt nun *export DISPLAY=firewall:portnumber*.

[2] IP-Spoofing = Eine Attacke, bei der ein fremdes System versucht die Identität eines anderen zu übernehmen, d.h. dessen IP-Adresse zu benutzen[7].

Anschließend kann er seine X-Applikation laufen lassen. Diese wird auf dem Rechner *hostname* angezeigt, wenn dort in einem POPUP-Window diese Verbindung zugelassen wird. D.h. jede neue X-Verbindung muß auf dem Rechner *hostname* bestätigt werden.

Das X-GW Gateway leitet somit X-Applikation transparent durch. Der interne Rechner braucht nur X-Verbindungen vom Firewall zuzulassen, da alle X-Verbindungswünsche von diesem zu kommen scheinen. Das POPUP-Window gewährleistet, daß nur gewünschte Verbindungen durch das Firewall durchgelassen werden.

4.7 login-sh, rlogin-gw und plug-gw

Als weitere Applikations-Dienste unterstützt das Firewall Toolkit eine Login-Shell, der ein Authentifizierungsschritt vorgeschaltet wird, ein rlogin-gw, welches rlogin Anfragen durchschaltet, wenn diese zugelassen wurden, sowie ein plug-gw. Dieses erlaubt es weitere TCP basierte verbindungsorientierte Applikationen, wie z.B. NNTP, transparent durchzuschalten.

Diese Dienste sollen bzw. brauchen im KFA-Umfeld nicht benutzt werden. Ihre Funktionalität wurde daher nicht überprüft.

4.8 Syslogd: System Logging

Ein spezieller Syslog-Daemon bietet die Möglichkeit, auf spezielle Logging-Einträge durch vorgegebene Programme (Trap, Mail, Beeper,...) zu reagieren, was Realtime-Scanning von Systemlogs erlaubt.

4.9 Authd: Netzwerk-Authentication Service

Im Allgemeinen sollte von außen weder auf den Firewall, noch auf lokale Rechner hinter dem Firewall zugegriffen werden, weil die Paßworte derzeit noch weitgehend unverschlüsselt übertragen werden, also durch Abhorchen des Netzes auszuspionieren sind. Die Kenntnis dieser Passwörter ermöglicht somit externen Benutzern den Zugriff auf interne Rechner und das Firewall. Andererseits müssen in manchen Situationen, z.B. reisenden Mitarbeitern, solche Zugriffe ermöglicht werden. Hier ist normaler Passwortschutz nicht ausreichend. One-Time-Passwords oder Challenge/Response Calculators müssen hier eingesetzt werden.

Hierzu bietet das Toolkit die Möglichkeit fünf unterschiedliche Authentifizierungs-Protokolle zu benutzen. Dies sind:

- normale Klartext–Paßworte
- Bellcore's S/Key [8][3]
- Digital Pathways SNK004 SecureNet KeyCards,
- Security Dynamics SecurID Cards [9] und
- Enigma Logic's Silver Card

Die Proxy-Server können so konfiguriert werden, daß eine Authentifizierung abhängig von der Quelle oder dem Ziel der Anwendung notwendig ist.

[3] S/Key ist verfügbar mittels FTP von: ftp://thumper.bellcore.com, in /pub/nmh/skey

5 Hardwarekomponenten und Netzlayout

Das TIS Firewall Toolkit läßt sich auf praktisch jedem durch den System-Administrator standardmäßig eingerichteten Unix-System installieren. Da das Zentralinstitut für Angewandte Mathematik für den Betrieb wichtiger Serversysteme bisher mit DEC Rechnersystemen gute Erfahrungen gemacht hat, wurde für die Testinstallation des Firewall Toolkits ebenfalls ein leistungsfähiger DEC–Alpha-Server gewählt. Die einzelnen Leistungsmerkmale des Firewall-Rechners sind:

System:	AlphaServer 1000 4/266 2 MB Cache 128 MB Hauptspeicher CDROM, Floppy 4 * 2.1 Gbyte Disk 2 * PCI-Fast-SCSI II 3 * PCI, 7 * EISA 2*seriell, 1 * parallel
Netzwerk-Anschlüsse:	2 * Ethernet DEC FDDIcontroller/EISA, DAS, Multimode Optics PCI-ATM-Adapter

Die hohe Gesamtleistung des Systems wird benötigt, da es den Netzverkehr möglichst transparent und ohne große Verzögerungen kontrollieren und gegebenenfalls weiterleiten soll. Ein extensives Logging und Accounting erfordert dementsprechend auch große Speicherkapazitäten und externe Platten. Security-Logging ist Langzeit-Logging, da Angriffe auf ein System nur konsequent verfolgt werden können, wenn auf Logging-Daten einer größeren Zeitspanne zurückgegriffen werden kann.

Den physikalischen Aufbau des Systems zeigt die Abbildung2

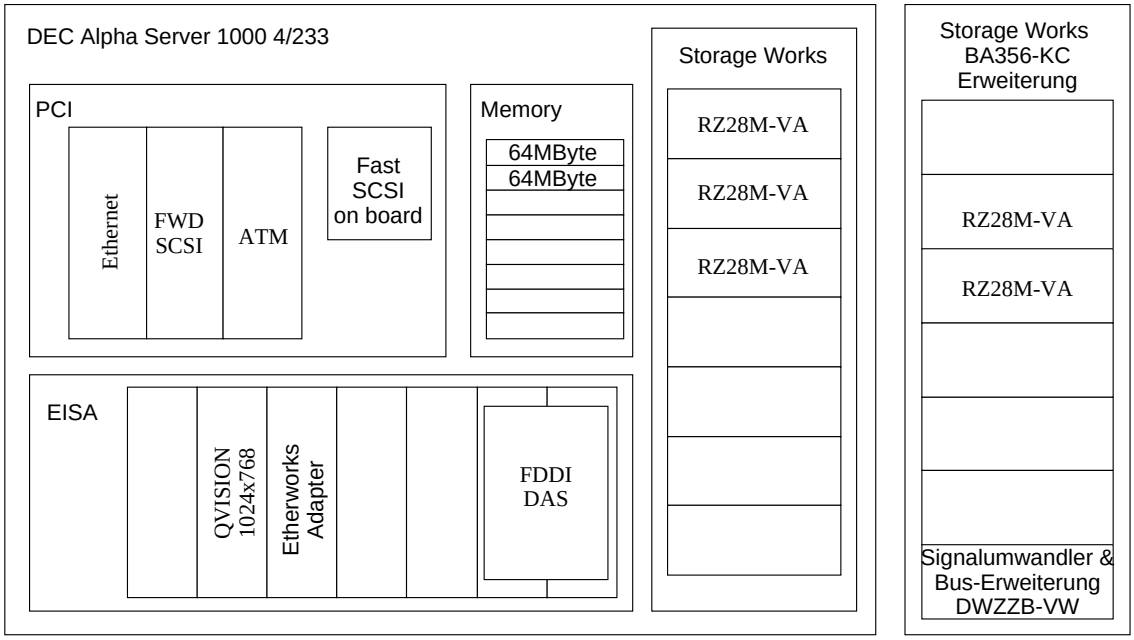

Abb. 2: Physikalischer Aufbau des Firewall-Systems

Das Netzlayout für das Testszenario mit dem KFAnet-Subnetz 134.94.108 als externem Test-Netz zeigt die Abbildung 3

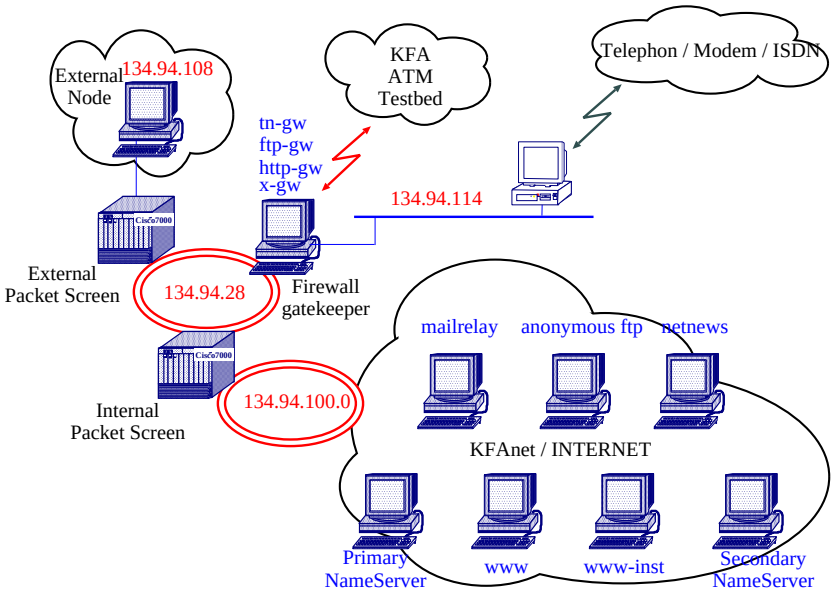

Abb. 3: Schematischer Netzaufbau

Als externer Rechner wurde eine IBM-RS6000/320 benutzt. Der *External Packet Screen* Router (Cisco7000) verhindert IP-Spoofing. Er ist dazu so konfiguriert, daß von außen keine Pakete mit Source IP-Adresse aus dem lokalen Netz 134.94.nnn, wobei nnn ¬= 108, über das externe Interface in das lokale Netz gelangen können. Der *Internal Packet Screen* Router (Cisco7000) wird mit den Access-Listen, wie in [5] spezifiziert konfiguriert. Zugriffe aus dem KFA-ATM-Testbed, sowie von externen Rechnern, die über Modem Zugang ins KFAnet/Internet erhalten möchten, können über das Firewall-Gateway geleitet werden, was einen ausreichenden Schutz gewährleistet.

Die im lokalen Netz gelegenen Server *Mailrelay*, *anonymous FTP*, *NetNews*, sowie *Primary Name*Server und *Secondary Name*Server stellen kein Sicherheitsrisiko dar, da sie ausreichend durch Access-Listen geschützt und nur mit den benötigten Diensten bestückt werden können.

Auf die Verlagerung des *HTTP*-Servers in den externen Bereich, vor das Firewall wurde in den Tests verzichtet. Aus Sicherheitsgründen wäre dies notwendig, da das HTTP-Protokoll bzw. ein HTTP-Server als unsicher angesehen werden muß. Ein sicherer HTTP-Server darf keine User-Accounts enthalten; auf ihn darf nur mittels gesicherter, evtl. verschlüsselter Protokolle zugegriffen werden und er darf nur die benötigten Dienste zur Verfügung stellen. Für interne Maschinen gilt er als externer Rechner, d.h. auch bei einem kompromittierten HTTP-Server dürfen interne Maschinen nicht angreifbar sein.

6 Installation, Konfiguration und Nutzbarkeit der einzelnen Dienste

Im Anschluß an die Installation, Konfiguration und den Netz-Anschluß des Firewall Rechners wurde die Firewall-Software auf dem Rechner installiert werden. Die vorgegebenen Defaultwerte der Installationsanleitung konnten weitgehend übernommen werden und sind, wo notwendig, an das System anzupassen.

Die nicht benötigten Dienste werden abgeschalten durch

- Editieren der Datei /etc/inetd.conf
- Editieren von Startup Scripts
- Editieren der Betriebssystem Konfiguration, um nichtgewünschte Kernel-basierte Netzwerk-Dienste auszuschalten
- Eventuell Erzeugen eines neuen Kernel

Anschließend wird Netacl konfiguriert, um begrenzten Zugriff auf den Firewall-Rechner von internen Systemen aus zu ermöglichen. Ebenso werden die Telnet und FTP-Gateway-Programme konfiguriert und durch Zugriffsrechte gesichert. Gleiches geschieht für das HTTP-Gateway sowie *smap* und *smapd*. Darauf folgend können Authentifizierungsprogramme eingearbeitet werden. Bei den vorliegenden Tests wurde jedoch nur mit Klartext-Paßworten die allgemeine Funktionalität der Authentifizierung überprüft. Die Funktionalität in Verbindung mit entsprechenden Key-Cards muß, wenn diese eingesetzt werden sollen, noch verifiziert werden.
Die Installation der einzelnen Dienste stellt kein Problem dar, da die mitgelieferten *Makefiles* nach entsprechender Anpassung an das Betriebssystem Digital Unix fehlerfrei funktionieren.

Die Konfiguration der einzelnen Dienste ist aufwendiger, aber aufgrund der mitgelieferten Dokumentationen leicht durchführbar.

Die Nutzung der Dienste ist teilweise gewöhnungsbedürftig und wird im folgenden in der Art einer Benutzereinführung dargestellt. Hierbei sei der Name des Firewall-Rechners im Folgenden: **gatekeeper.KFA-Juelich.de**.
System-Prompt erscheint in *Italic*, feststehende Strings in **Bold**, beliebige Benutzereingabe in ***Bold-Italic***

6.1 E-Mail

Das allgemeine Firewall-Konzept sieht Mail-Zugang von außen grundsätzlich über offizielle Mailadressen vor. Diese werden auf dem Firewall-Rechner *gatekeeper* in der Datei /etc/aliases eingetragen und verweisen auf die realen Mailadressen der internen Rechner. *gatekeeper* arbeitet somit als transparentes Applikation Gateway mit dem Internet-Namen *KFA-Juelich.de* bzw. *mailrelay.KFA-Juelich.de*. Auf *gatekeeper* wird die ankommende Mail nicht vom *sendmail*-Prozess, sondern von dem vorgelagerten Prozess *smap* empfangen und über den Prozess *smapd* an den *sendmail*–Prozess weitergeleitet.

Die Installation der Programme *smap* und *smapd* lief ohne Probleme ab. Der automatisch beim Systemboot gestartete Prozess *sendmail* muß aus der Konfiguration entfernt werden. Dies geschieht durch Umbenennen des Scripts *S40sendmail* im Directory */sbin/rc3.d* in z.B. *no_S40sendmail* und Beendigen von sendmail mittels */sbin/inet.d/sendmail stop* oder anschließendem Systemreboot.

Für den externen Mailbenutzer sind keine Besonderheiten zu beachten. Das Mail-Applikation-Gateway arbeitet konform zu einem normalen Mail-Server. Es müssen somit keine besonderen oder zusätzlichen Kommandos ausgeführt werden.

Mail nach extern wird mittels *direct delivery* grundsätzlich sofort an den entfernten Rechner geleitet.

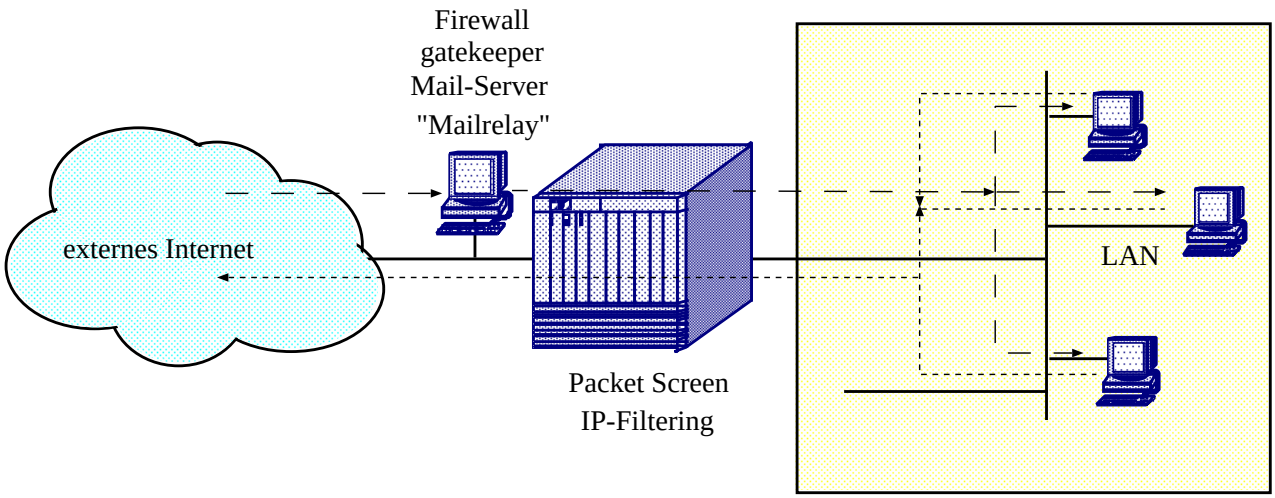

Abb. 4: mailrelay.zam.kfa-juelich.de

6.2 FTP

Der FTP-Proxy-Service erfordert vom lokalen Benutzer zuerst eine Verbindung zum lokalen Firewall-Gateway:

user@myhost> **ftp gatekeeper.KFA-Juelich.de**

Will man nun als anonymous-User eine Datei auf dem Rechner ftp.nirgendwo.net per FTP abholen, so muß man als Username ***anonymous@ftp.nirgendwo.net*** angeben. Eine Kommandosequenz hat dann die folgende Form:

user@myhost> **ftp gatekeeper.KFA-Juelich.de**
Connected to gatekeeper.zam.kfa-juelich.de.
220– gatekeeper FTP Proxy (Version 1.3) ready.
Name (gatekeeper:you): ***anonymous@ftp.nirgendwo.net***
331– (—GATEWAY CONNECTED TO ftp.nirgendwo.net —)
331– (220 ftp.nirgendwo.net FTP server (Version 4.1 Mon Mar 17:05:29 CST 1996) ready
331 Guest login ok, send ident as password.
Password:########
230 Guest login ok, access restriction apply.
ftp> ...

Ist man einmal mit dem entfernten System verbunden, so werden alle folgenden Kommandos transparent an das entfernte System weitergeleitet. Wurde das Passwort falsch angegeben, so kann normal weiterverfahren werden, als ob das Gateway nicht existieren würde.

Als externer Benutzer, der per FTP auf einen lokalen Rechner zugreifen will, muß man sich zuerst Authentifizieren. Hierzu wird das FTP user Kommando zweifach benutzt:

user@extern_host> **ftp gatekeeper.KFA-Juelich.de**
Connected to gatekeeper.zam.kfa-juelich.de.
220–Before using the proxy you must first authenticate
220– gatekeeper FTP Proxy (Version 1.3) ready.
Name (gatekeeper:you): ***it_is_me***
331 Enter authentication password for it_is_me
Password:########
230 User authenticated to proxy
ftp> ***local_user@local_host.inst.KFA-Juelich.de***
331– (—GATEWAY CONNECTED TO —)
331– (220 local_host.inst.KFA-Juelich.de FTP server (Version 3.2 Mon Mar 17:05:29 CST 1996) ready
331 Password required for local_user.
Password:########
230 User local_user logged in.
ftp> ...

Der erste FTP User Aufruf authentifiziert den Benutzer dem Firewall gegenüber als berechtigter Benutzer des Firewall-FTP-Gateways. Der zweite Aufruf authentifiziert den Benutzer dem lokalen System gegenüber.

Das folgende Bild verdeutlicht den Datentransfer für FTP-Gateway-Verbindungen:

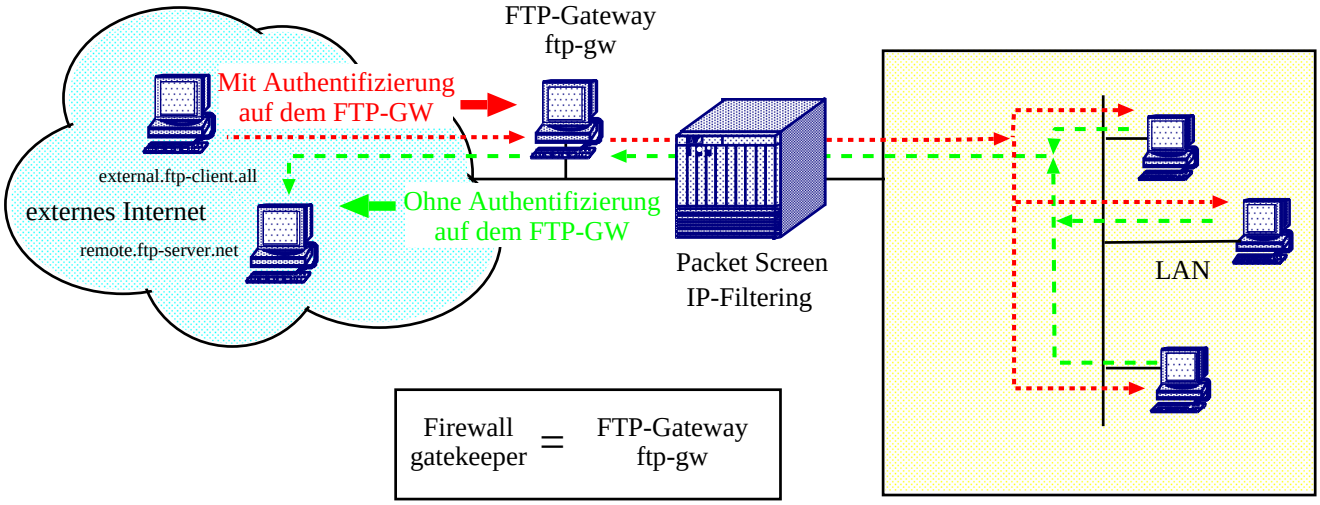

Abb. 5: ftp-gw.zam.kfa-juelich.de

6.3 Telnet

Die Benutzung des Telnet–Gateways geschieht in gleicher Weise, indem von externer Seite eine Verbindung zum Firewall aufgebaut wird. In die andere Richtung ist der Telnet-Zugriff von lokalen Rechnern auf externe Rechner ohne Beschränkung erlaubt, da das Firewall nur lokale Rechner zu schützen hat.

Nach der Authentifizierung auf dem Firewall-Rechner muß der externe Benutzer mittels **connect *hostname*** eine transparente Verbindung vom Telnet-Gateway zum lokalen Rechner ***hostname*** aufbauen.

Eine entsprechende Kommandosequenz hat dann etwa die folgende Form:

user@extern> **telnet gatekeeper.KFA-Juelich.de**
Trying 134.94.28.4 ...
Connected to gatekeeper.zam.kfa-juelich.de.
Escape character is '^]'
gatekeeper telnet proxy (Version 1.3) ready.
Before using the proxy you must first authenticate
Username: ***it_is_me***
Password: ***########***
Login Accepted
tn—gw—> **connect *telnet-host.inst.KFA-Juelich.de***
Digital Unix (telnet-host.inst.KFA-Juelich.de) (ttyp2)

*login:****my_account***
Password########
Last login: Wed Jun 19 12:09:06 from anyhost
my_account@telnet-host.inst.KFA-Juelich.de> ...

Allgemein sollte die Authentifizierung allerdings nicht mit Klartext-Paßworten durchgeführt werden, sondern zumindest mit One-Time-Passwords oder entsprechenden Protokollen (S/Key [8], Hand Held Authenticators [9]...). Eine Verschlüsselung der gesamten Session ist empfehlenswert und kann z.B. mit dem Produkt STEL [10] durchgeführt werden. Auf eine Untersuchung dieser Authentifizierungsmöglichkeiten wird jedoch in diesem Bericht verzichtet.

Mittels STEL kann bei Eintragung eines Benutzers ein erstes Anfangspasswort eingetragen werden. Zukünftige Logins von entfernter Stelle werden immer verschlüsselt durchgeführt, demzufolge kann der Benutzer auch über das Netz hinweg sein Passwort sicher ändern. Häufiges Ändern des auf dem KFA-internen Rechner gespeicherten *Security-Keys* erhöht die Sicherheit der Session zusätzlich. Ein wesentlicher Vorteil des STEL in Verbindung mit einem TELNET-Gateway liegt darin, daß über die verschlüsselte Session zum Gateway hin nun KFA-interne UserId's und Passwörter für interne Telnet-Sessions übertragen werden können. (Diese sind, da sie Teil des Datastream der Telnet-Gateway Session sind, ebenfalls verschlüsselt.)

Das folgende Bild verdeutlicht den Datentransfer für Telnet-Verbindungen:

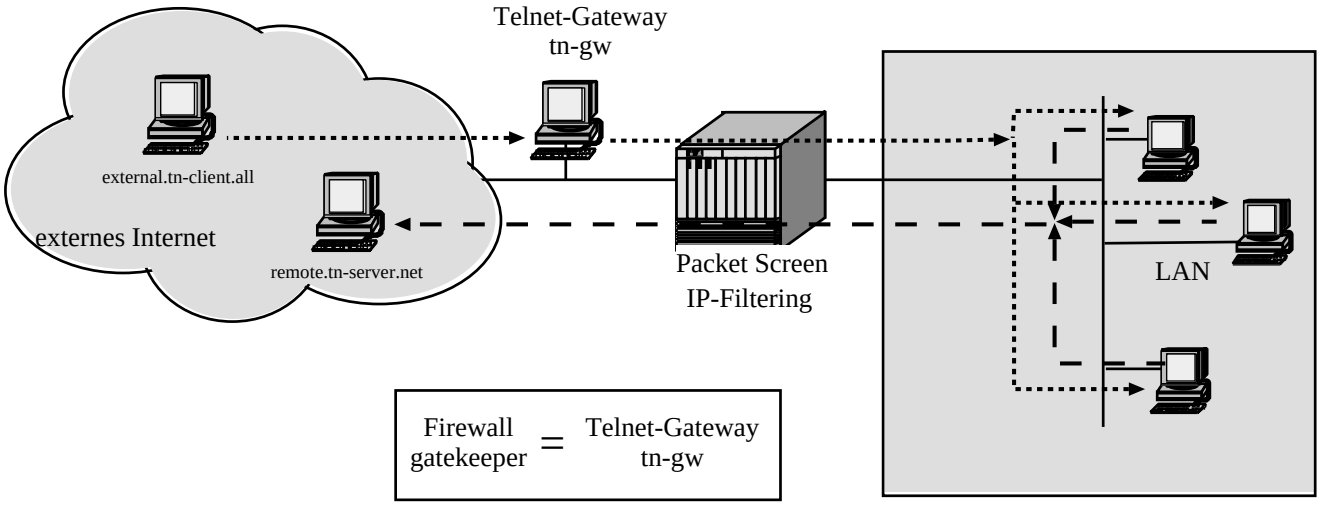

Abb. 6: tn-gw.zam.kfa-juelich.de

6.4 WWW

Das Firewall Toolkit bietet mit dem HTTP-GW Gateway die Möglichkeit WorldWideWeb-Applikationen über das Firewall hinweg zu benutzen.

Geht man davon aus, daß WWW Zugriffe von außen auf lokale WWW-Server und von lokalen WWW-Clients nach außen durch die *Internal Packet Screen* unterbunden werden, so müssen interne Benutzer ihre WWW-Anforderungen über das HTTP-GW-Gateway befriedigen. Hierzu müssen sie nur vor der ursprünglichen URL den String http://firewall/ anfügen. Das Gateway leitet die Requests dann transparent weiter. Will man z.B. den WWW Server des DFN Network Operation Centers erreichen, so spezifiziert man:

http://gatekeeper.KFA-Juelich.de/*http://wwwnoc.dfn.de*

Durch Einstellungen auf dem HTTP-GW-Server können für unterschiedliche Rechner unterschiedliche Default-Server spezifiziert werden. So könnte zum Beispiel mit **http://gatekeeper.KFA-Juelich.de** auf dem Rechner *x.inst1.KFA-Juelich.de* als Einstieg die WWW-Seite des Institutes *inst1* des Forschungszentrums erreichbar sein, während auf dem Rechner *y.inst2.KFA-Juelich.de* als Einstiegs-Seite die WWW-Seite des Forschungszentrums Jülich erscheint. Wird in den WWW-Client-Applikationen, in denen das möglich ist, als Proxy-Server der Firewall-Rechner eingetragen, so ist bei diesen keine Modifikation (voranstellen von *http://gatekeeper.KFA-Juelich.de/*) erforderlich.

Weiterhin können auch hier, genau wie bei FTP und Telnet, entsprechende Rechner berechtigt werden, den Service zu nutzen, während andere dies nicht dürfen.

Bezüglich der allgemeinen Sicherheit von WWW-Servern vergleiche man [11].
Das logische Schaubild hierzu hat die folgende Form:

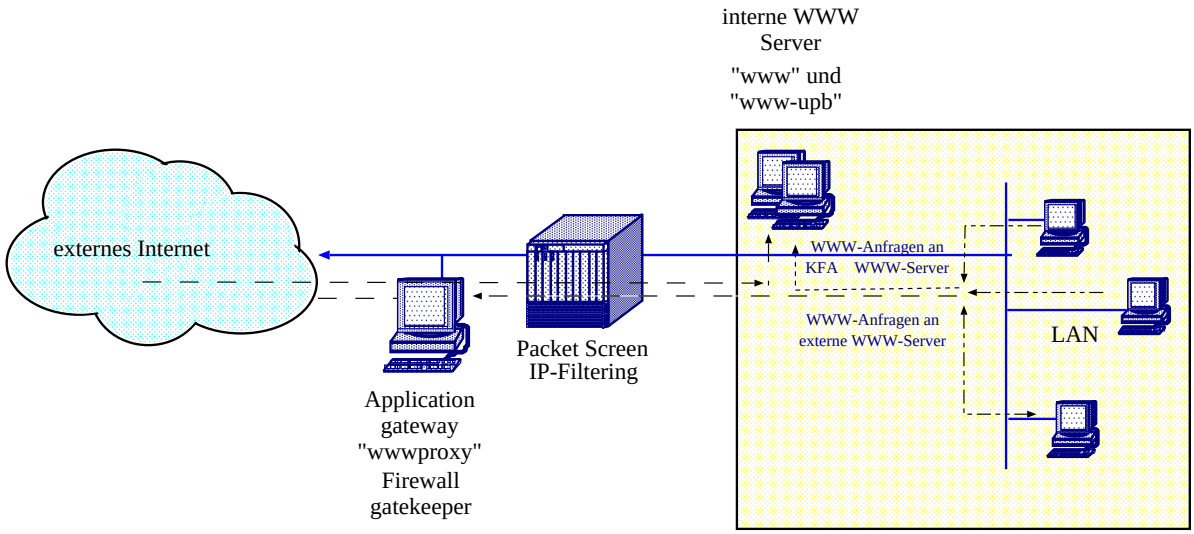

Abb. 7: WWW-(W3)-Service

6.5 X-Protocol

Das X-Window System ist eine Netzwerk-transparente graphische Benutzer-Interface–Technologie für bitmapped Displays. Es ist eine Sammlung von Protokoll-Definitionen, File-Formaten, Dokumentationen und einfachen Software-Quell-Programmen in C für Server, Klienten und Utility-Programme.

X unterscheidet sich von anderen graphischen Benutzeroberflächen, indem die graphische Funktionalität in einen Client- und Server-Teil aufgespalten wird. Der X-Server-Teil erhält exclusive Kontrolle über den Bildschirm und verteilt Anforderungen an die Clients.

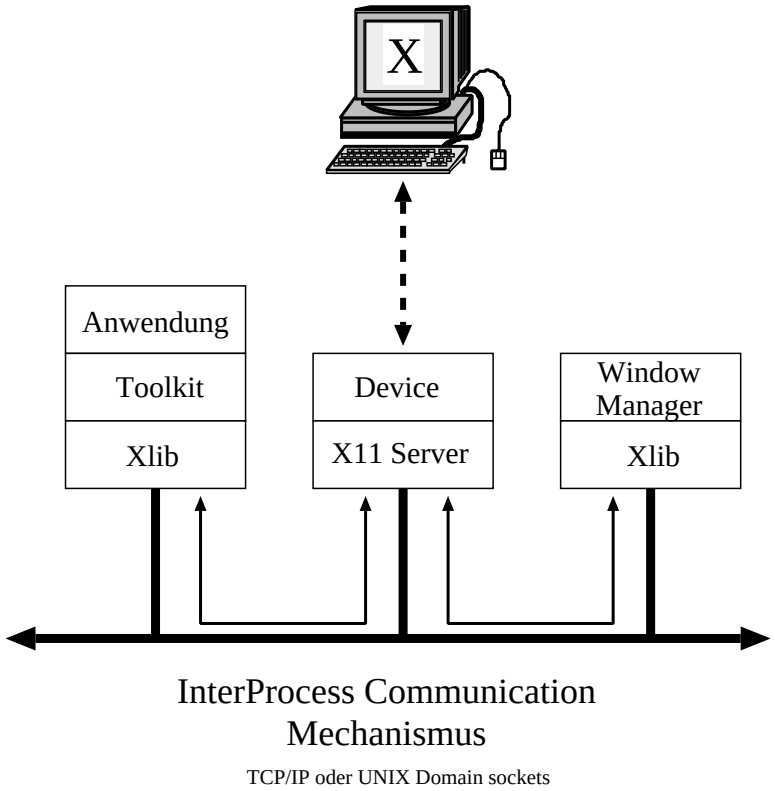

Abb. 8: X Window System Architektur

Da der X-Server verschiedenen Client-Anwendungen Zugriff auf eine gemeinsame Ressource, den Bildschirm, erlaubt, ergibt sich ein potentieller Konflikt.

Server- und Eingabe-Aktivitäten werden den X-Anwendungen als Nachrichten, sogenannte *events*, übertragen. Wenn ein Window für eine Anwendung eröffnet wird, wird in eine Liste eingetragen, über welche *events* diese Anwendung unterrichtet werden soll. Steht Eingabe oder Server-Aktivität an, überprüft der Server, an welche X-Anwendungen diese geleitet werden sollen. Obwohl normalerweise ein Client nur über *events* im eigenen Window informiert wird, kann er verlangen, über alle *events* informiert zu werden. Dies bedeutet, daß er alle Eingaben, also auch *login requests* und Passwörter mitlesen kann. Dies bedeutet ein großes Sicherheitsrisiko.

Das Basis-Sicherheitsmodell für X-Windows erlaubt dem Benutzer, die Menge der Rechner zu bestimmen, die Verbindungen zu seinem X-Server aufnehmen dürfen. Beim Firewall Toolkit wird die Menge der erlaubten Verbindungen dahingehend eingeschränkt, daß der lokale Rechner nur noch vorher authentisierte Verbindungswünsche vom Firewall aus zuläßt.

Der Ansatz der Nutzung des MIT-MAGIC-COOKIE kann, wie in [5] erläutert, aufgrund der unverschlüsselten Übertragung des Magic-Cookie nicht für externe Kommunikation genutzt werden und stellt somit keine Alternative dar.

Der rechnerbasierte Ansatz des X-Window-Systems wird beim X-GW Programm des Firewall Toolkits durch einen applikationsbezogenen Ansatz ersetzt. Jeder Aufruf einer X-Applikation wird einzeln bestätigt und genehmigt.

Vereinfacht gesagt bietet X-GW eine X-Verbindung auf Benutzer-Ebene unter der Kontrolle des tn-gw Telnet-Gateways. Client-Applikationen auf einem beliebigen Rechner starten ihre X-Anwendung auf einem virtuellen Display des Firewall Rechners. Wenn der Request beim Firewall-Rechner ankommt, generiert dieser ein POPUP-Fenster auf dem tatsächlich gewünschten X-Display auf einem lokalen Rechner, in dem der lokale Benutzer aufgefordert wird, den Verbindungswunsch zu bestätigen. Bei jedem erneuten Aufruf einer X-Applikation wird dann ein neues POPUP-Fenster generiert, das der lokale Benutzer zu bestätigen hat. Die virtuelle Display-Nummer des Firewall Rechners erhält der Benutzer durch den Aufruf einer Telnet Session:

user@myhost> **telnet gatekeeper.KFA-Juelich.de**
Trying 134.94.28.4 ...
Connected to gatekeeper.zam.kfa-juelich.de.
Escape character is '^]'
gatekeeper telnet proxy (Version 1.3) ready.
Before using the proxy you must first authenticate
Username: ***it_is_me***
Password: ********
Login Accepted
tn—gw—> **x *hostname***
tn-gw—> display port=gatekeeper.zam.kfa-juelich.de:10
tn-gw —> **exit**
Disconnecting
Connection closed by foreign host.

Setzt er nun in seiner Shell **export DISPLAY=gatekeeper.zam.kfa-juelich.de:10**, so kann er X-Applikationen auf dem entfernten Rechner anzeigen lassen, sofern der lokale Benutzer dies jeweils zugelassen hat.

Direkte X-Verbindungen zwischen lokalem Rechner und entferntem Rechner brauchen bei diesem Protokoll nicht auf der Packet Screen durchgelassen zu werden.

7 Zusammenfassung der Untersuchung

Die untersuchten Softwarekomponenten des Firewall Toolkit lassen sich einfach und komfortabel installieren. Anpassungen bezüglich des Betriebssystems sind nur an wenigen Stellen erforderlich und in zwei vor der Installation anzupassenden Initialisierungsteilen, *Makefile* und *firewall.h*, zusammengefaßt.

Die Defaulteinstellungen geben die wesentlichen Optionen vor. Für nicht konforme Systeme sind die gängigen Alternativen als Kommentare beigefügt.

Die Gesamtsoftware benötigt nur wenige Änderungen in den Betriebssystem-Komponenten. Änderungen sind im Wesentlichen in */etc/inetd.conf* und */etc/services* durchzuführen. Die Konfiguration der einzelnen Dienste, im Besonderen für die Zugangsberechtigung über Internet-Adressen (Access-Listen) sowie für besondere Optionen bei einzelnen Diensten (Authentifizierung, Logging, Read-Only bei FTP, ...), wird in einer einzigen Datei */usr/local/etc/netperm-table* vorgenommen. Dies ermöglicht eine überschaubare Installation der Dienste, wobei jeder Dienst einzeln konfigurierbar ist.

Das Baukastenprinzip der einzelnen Dienste erlaubt es, wenn dies erforderlich ist, Teile des Toolkits auch an beliebigen Stellen im lokalen Netz ohne ein allgemeines Firewall zu installieren. Auf diese Weise können besonders schützenswerte Rechner zusätzlich gesichert werden.

Bei der Installation und dem Test des Toolkits sollten die Syslog-Dateien überwacht werden, da einige der Softwarekomponenten hier ihre Fehler-Meldungen ablegen.

Grundsätzlich sollten, und so ist es auch in der Installationsanweisung nachzulesen, Systemupgrades nur dann vorgenommen werden, wenn dies notwendig ist. *"If it isn't broken, don't fix it" is the best policy* [12]. System-Upgrades auf dem Firewall-Rechner sollten nur vorgenommen werden, wenn Bugs zu beheben sind oder wenn neue oder zusätzliche Funktionen gewünscht werden.

Der Managementaufwand eines Firewallrechners entspricht dem eines normalen Unix-Systems. Da keine Benutzer–Accounts auf dem System existieren sollten und da System-Updates nur dann installiert werden, wenn ansonsten die Sicherheit gefährdet wäre, ist der Aufwand sogar geringer als bei einem normalen Unix-System. Hauptaufgaben des Systemadministrators bestehen in der Kontrolle der System-Dateien, sowie in der Überwachung der File-Systeme, da Logging und Accounting große Datenmengen produzieren und somit die File-Systeme zum Überlaufen bringen können.

Das Toolkit enthält weitere Reporting-Programme, die zusätzlich installiert und für automatisch generierte Verkehrsstatistiken genutzt werden können. Allgemein wird empfohlen, eine wöchentliche Statistik über Firewall-Aktivitäten und eine nächtliche Statistik über sicherheitsrelevante Aktivitäten zu generieren und als Mail an den Firewall-Administrator zu versenden. Diese Tools wurden nicht untersucht.

Aus Benutzersicht ist das Firewall Toolkit ein wenig gewöhnungsbedürftig, da zum Teil zusätzliche Schritte durchgeführt werden müssen (mehrfaches Authentisieren: beim Firewall und beim Zielrechner, zusätzliches Aktivieren einer Telnet-Session zum Firewall hin für die Authentisierung und Ermittlung des virtuellen Displays). Nach einer kurzen Eingewöhnungsphase stellt dies jedoch kein Problem mehr dar.

Zusammenfassend kann das Firewall Toolkit als ein gelungenes, frei verfügbares Programmpaket zur selbständigen Installation und Konfiguration angesehen werden.

Durchsatzbetrachtungen wurden nicht durchgeführt, da die zu erwartenden Datenströme nur sehr schwer simulierbar sind. Verändertes Benutzerverhalten aufgrund neuer Software-Produkte ändert kurzfristig und unvorhersehbar die Nutzung der Firewall-Komponenten. Entgültige Aussagen über die Verwendbarkeit in einer Netz-Umgebung mit Durchsatzraten von 34, 155 oder z.B. 622 Mb/s zu externen Partnern lassen sich somit nicht machen. Für einen lokalen Bereich, der abgetrennt vom Gesamtnetz gesichert werden soll, stellt das Toolkit eine gute Möglichkeit dar, Internet-Konnektivität aufrechtzuerhalten bei gleichzeitiger Abschottung vor nicht erwünschten Besuchern.

8 Anhang

8.1 Modifizierte und zusätzlich notwendige Dateien

/usr/local/etc/netperm-table

```
#
# Rechner 134.94.16.65 erhaelt Zugriff auf den Firewall mittels Telnet
netacl-telnetd:   permit-hosts 134.94.16.65 -exec /usr/sbin/telnetd
# Rechner aus dem Netz 134.94.108 und der Rechner 134.94.16.135 duerfen
# das tn-gw Gateway benutzen.
netacl-telnetd:   permit-hosts 134.94.108.* -exec /usr/local/etc/tn-gw
netacl-telnetd:   permit-hosts 134.94.16.135 -exec /usr/local/etc/tn-gw
# Alle anderen duerfen das tn-gw Gateway nicht benutzen
netacl-telnetd:   deny-hosts * -exec /usr/local/etc/tn-gw
# -------------------
# Smap laeuft unter der Userid mit der UID 6
smap, smapd:      userid 6
# smap und smapd directory fuer die Ablage der Mails ist /var/spool/smap
smap, smapd:      directory /var/spool/smap
# smapd ist im Dateisystem unter /usr/local/etc/smapd zu finden
smapd:            executable /usr/local/etc/smapd
smapd:            sendmail /usr/sbin/sendmail
smap:             timeout 3600
# -------------------------
# Zur Authentifizierung wird der AUthentication Server auf Port 7777
# benutzt
ftp-gw:           authserver localhost 7777
# Dateien in denen Messages stehen sind ftp-deny.txt, ftp-welcome.txt
# und ftp-help.txt
ftp-gw:           denial-msg /usr/local/etc/ftp-deny.txt
ftp-gw:           welcome-msg /usr/local/etc/ftp-welcome.txt
ftp-gw:           help-msg /usr/local/etc/ftp-help.txt
ftp-gw:           timeout 3600
# Rechner 134.94.16.65 darf ohne zusaetzliche Authorisierung das
# FTP-Gateway benutzen
ftp-gw:           permit-hosts 134.94.16.65
# Rechner aus dem Netz 134.94.108 duerfen nach entsprechender
# Authentifizierung ueber den Authentification Server das FTP-Gateway
# benutzen
ftp-gw:           permit-hosts 134.94.108.* -authall
# Alle anderen Rechner duerfen das FTP-Gateway nicht benutzen
ftp-gw:           deny-hosts *
# ---------------------------
# Dateien in denen Messages stehen sind tn-deny.txt, tn-welcome.txt
# und tn-help.txt
tn-gw:            denial-msg /usr/local/etc/tn-deny.txt
tn-gw:            welcome-msg /usr/local/etc/tn-welcome.txt
tn-gw:            help-msg /usr/local/etc/tn-help.txt
tn-gw:            timeout 3600
# Rechner aus dem Netz 134.94.108 duerfen das tn-gw Gateway benutzen,
# wenn sie sich authentifiziert haben. Ebenfalls duerfen sie den
# integrierten X-Server x-gw benutzen
tn-gw:            permit-hosts 134.94.108.* -auth -xok
# Rechner 134.94.16.135 darf das tn-gw Gateway benutzen
tn-gw:            permit-hosts 134.94.16.135
# Alle anderen Rechner duerfen das tn-gw Gateway nicht benutzen
tn-gw:            deny-hosts *
# ---------------------------
# Meldet sich jemand beim WWW-Server mit http://gatekeeper.KFA-Juelich.de,
```

```
# so wird er automatisch an den KFA-WWW-Server weitergeleitet
http-gw:          default-httpd www.kfa-juelich.de
# Rechner 134.94.16.65 darf das http-gw Gateway benutzen

http-gw:          permit-hosts 134.94.16.65
# Rechner 134.94.16.65 darf das http-gw Gateway benutzen, und wird
# automatisch an den WWW-Server hlrz31.hlrz.kfa-juelich.de weitergeleitet.
http-gw:          permit-hosts 134.94.16.135 -httpd hlrz31.hlrz.kfa-juelich.de
# Alle anderen Rechner duerfen das HTTP-Gateway nicht benutzen
http-gw:          deny-hosts *

# Example auth server and client rules
# -----------------------------------
# Die lokale Datenbasis wird fuer den Authentifizierungsprozess benutzt.
# Den Authentifizierungsserver darf nur der lokale Rechner, das Firewall
# selbst, benutzen.
authsrv:          database /var/adm/authsrv.db
authsrv:          permit-hosts localhost
# clients using the auth server
*:                authserver 127.0.0.1 7777
# X-forwarder rules
# Zur Benutzung des X-GW gateways koennen sowohl das tn-gw als auch
# das rlogin-gw benutzt werden
tn-gw, rlogin-gw: xforwarder /usr/local/etc/x-gw
```

/etc/services

```
#
# ************************************************************
# *                                                          *
# *  Copyright (c) Digital Equipment Corporation, 1991, 1995   *
# *                                                          *
# * All Rights Reserved.  Unpublished rights  reserved  under  *
# .....                                                      *
# HISTORY                                                    *
#                                                            *
# @(#)$RCSfile:services,v $ $Revision:4.2.13.3 $ (DEC)       *
#                           $Date:1994/02/03 21:43:31 $      *
#                                                            *
# Description:The services file lists the sockets and        *
#             protocols used for Internet services.          *
#                                                            *
# Syntax:                                                    *
#   ServiceName PortNo./ProtName [ali_1,...,ali_n] [#comments] *
#                                                            *
# ServiceName  official Internet service name                *
# PortNumber   the socket port number used for the service   *
# ProtocolName the transport protocol used for the service   *
# alias        unofficial service names                      *
# #comments    text following the comment char (#) is ignored  *
# ************************************************************
#
echo        7/tcp
echo        7/udp
discard     9/tcp    sink null
discard     9/udp    sink null
systat     11/udp    users
daytime    13/tcp
daytime    13/udp
netstat    15/tcp
quote      17/udp    text
chargen    19/tcp    ttytst
chargen    19/udp    ttytst
ftp        21/tcp
# Zusaetzlicher Eintrag fuer den FTP-Gateway-Prozess
ftp-gw     21/tcp
telnet     23/tcp
# Zusaetzlicher Eintrag fuer den Telnet-Gateway-Prozess
tn-gw      23/tcp
# Zusaetzlicher Eintrag fuer den Telnet-Prozess auf
# den Firewall-Server (Durch netacl auf bestimmten Rechner
# beschraenkt)
telnet-a   24/tcp
smtp       25/tcp    mail
time       37/tcp    timserver
time       37/udp    timserver
name       42/tcp    nameserver
.....
# Zusaetzlicher Eintrag fuer den HTTP- bzw WWW-Gateway-Prozess
http-gw    80/tcp
.....
# Zusaetzlicher Eintrag fuer den Authentication-Server
authsrv    7777/tcp
.....
```

/etc/inetd.conf

```
#
# ************************************************************
# *                                                          *
# * Copyright (c) Digital Equipment Corporation, 1991, 1992  *
# *                                                          *
# * All Rights Reserved.  Unpublished rights  reserved  under  *
# .....                                                      *
# ************************************************************
#                                                            *
# Internet server configuration database                     *
#                                                            *
# Description: The inetd.conf file is the file that the inetd  *
#              daemon reads for information on how to handle   *
#              Internet service requests.                      *
#                                                            *
# Syntax: ServiceName  SocketType  ProtocolName  Wait/NoWait \ *
#                UserName  ServerPath  ServerArgs              *
#                                                            *
# ServiceName   name of an Internet service defined in the     *
#                  /etc/services file                        *
# SockettType   type of socket used by the service, either     *
#                  stream or dgram                           *
# ProtocolName  name of an internet protocol defined in the    *
#                  /etc/protocols file                       *
# Wait/NoWait   determines whether the inetd daemon waits for  *
#                  a datagram server to release the socket   *
#                  before continuing to listen at the socket   *
# UserName      the login that inetd should use to start the   *
#                  server                                    *
# ServerPath    full pathname of the server                  *
# ServerArgs    optional command line arguments that inetd     *
#               should use to execute the server             *
#                                                            *
# ************************************************************
#
# Umlenkung des FTP-Streams ueber das NETACL-Programm
ftp       stream  tcp     nowait  root /usr/local/etc/netacl  in.ftpd
# Zusaetzlicher Eintrag fuer den FTP-Gateway-Prozess
ftp-gw    stream  tcp     nowait  root /usr/local/etc/ftp-gw  ftp-gw
# Umlenkung des ueber speziellen Port erreichbaren Telnet-Daemons ueber
# das  NETACL-Programm
telnet-a  stream  tcp     nowait  root /usr/local/etc/netacl  in.telnetd
# Zusaetzlicher Eintrag fuer den Telnet-Gateway-Prozess
telnet    stream  tcp     nowait  root /usr/local/etc/tn-gw   tn-gw
# Aenderung des SMTP-Dienstes: Nihct mehr ueber sendmail, sondern
# ueber smap-Dienstprogramm
smtp      stream  tcp     nowait  root /usr/local/etc/smap    smap
# Nutzung des HTTP-Proxy-Servers fuer Zugriffe ueber den Firewall hinweg
http-gw   stream  tcp     nowait  root /usr/local/etc/http-gw http-gw
#
comsat    dgram   udp     wait    root /usr/sbin/comsat       comsat
time      dgram   udp     wait    root internal               time
echo      stream  tcp     nowait  root internal               echo
echo      dgram   udp     wait    root internal               echo
discard   stream  tcp     nowait  root internal               discard
discard   dgram   udp     wait    root internal               discard
```

8.2 TIS Internet Firewall Toolkit License Agreement[4]

TRUSTED INFORMATION SYSTEMS, INC
TIS INTERNET FIREWALL TOOLKIT LICENSE AGREEMENT

February 9, 1996

Trusted Information Systems,Inc.(TIS)has developed the TIS Internet Firewall Toolkit (FWTK), a software kit for building and maintaining internetwork firewalls. The FWTK is distributed in source code form, with all modules written in the C programming language and runs on many BSD UNIX derived platforms. The TIS Internet Firewall Toolkit is a product of Trusted Information Systems, Inc. and is being made available for use on the following basis.

TIS grants you a license as follows to the TIS Internet Firewall Toolkit programs:

1. LICENSE. TIS grants you a non-exclusive, non-transferable license for the TIS Firewall Toolkit programs (the "FWTK") and its associated documentation, subject to all of the following terms and conditions. In accepting a copy of the FWTK you agree to the following terms and conditions.

This license permits you to use, copy, and modify the FWTK solely for your organization's use.

2. LIMITATIONS ON LICENSE.

a. You may only use, copy, and modify the FWTK as expressly provided for in this Agreement. You must reproduce and include this Agreement, and TIS's copyright notices on any copy and its associated documentation.

b. No part of the FWTK may be incorporated into any program or other product that is sold, or for which any revenue is received without written permission of Trusted Information Systems, Inc. A commercial license will be required in this case.

c. A person or organization may not provide, configure, install, or build the FWTK for a client or customer under any circumstances without the prior written consent of TIS. Such consent shall not be unreasonably withheld for a maximum of two (2) installations or configurations in a calendar year. Should TIS grant such consent, the provider must clearly state in documentation and bid/proposal materials that the TIS Internet Firewall Toolkit

[4] Entnommen aus: [3]

technologies are licensed and provided by Trusted Information Systems, and a copy of this license must be included with the configured system.

d. The FWTK, if modified, must carry prominent notices stating that changes have been made, and the dates of any such changes.

e. All rights not expressly granted herein are reserved to TIS.

3. NO TIS OBLIGATION: You are solely responsible for maintaining the FWTK and the security of the operating environment in which the FWTK may be used. You are solely responsible for all of your costs and expenses incurred in connection with the distribution of the FWTK or any application program hereunder, and TIS shall have no liability, obligation or responsibility therefor. TIS shall have no obligation to provide maintenance, support, upgrades, or new releases to you.

4. NO WARRANTY OF PERFORMANCE. The FWTK and its associated documentation are licensed "as is" without warranty as to their performance, merchantability, or fitness for any particular purpose. The entire risk as to the results and performance of the FWTK is assumed by you. Should the FWTK prove defective, you assume the entire cost of all necessary servicing, repair, or correction.

5. LIMITATION OF LIABILITY. Neither TIS nor any other person who has been involved in the creation, production or delivery of the FWTK shall be liable to you or to any other person for any direct, indirect, special, incidental, consequential, or punitive damages, even if TIS has been advised of the possibility of such damages.

6. TERM. The license granted hereunder is effective until terminated. This license shall automatically terminate without notice if you breach any of the provisions hereof. You may terminate it at any time by destroying the FWTK, any compiled object code derived therefrom, and its associated documentation.

7. GENERAL.

a. This Agreement shall be governed by the laws of the State of Maryland.

b. Address all correspondence regarding this license to TIS’s electronic mail address <fwtk-license@tis.com>, or to

Trusted Information Systems
2277 Research Blvd.
Rockville, Maryland 20850

8.3 TIS Internet Firewall Toolkit — Disclaimer[5]

TIS provides this software as a public service to experienced systems administrators who wish to build Internet firewalls. While we have made every effort to adequately document, comment, and package this software to make it as easy to use and install as possible, we do not recommend that it be used by persons unfamiliar with UNIX and UNIX security.

The firewall toolkit is intended to be useable on a wide range of UNIX systems. In order to keep it portable, neither the software nor the documentation delves into the specifics of different UNIX operating systems. The toolkit is security software, and relies heavily on features of the base operating system and the UNIX software model. Installers who are not very familiar with the details of their version of UNIX and its particular security problems may not be able to install the toolkit securely. TIS assumes no responsibility and cannot afford to support users who have questions about the basics of their systems. In a very real sense, this software is intended for use by security experts and experienced system administrators only.

Our support policy with respect to the freely available version of this software is that we will do our best to fix bugs in a timely manner, and we are more than happy to try to work with you to resolve any bugs in our software. We cannot, however, provide individual support on basic security issues or address specific operating system bugs or incompatibilities.

[5] Entnommen aus: [3]

8.4 Stichwortverzeichnis[6]

Glossary of Terms

Abuse of Privilege	When a user performs an action that they should not have, according to organizational policy or law.
Application-Level Firewall	A firewall system in which service is provided by processes that maintain complete TCP connection state and sequencing. Application level firewalls often re-address traffic so that outgoing traffic appears to have originated from the firewall, rather than the internal host.
Authentication	The process of determining the identity of a user that is attempting to access a system.
Authentication Token	A portable device used for authenticating a user. Authentication tokens operate by challenge/response, time-based code sequences, or other techniques. This may include paper-based lists of one-time passwords.
Authorization	The process of determining what types of activities are permitted. Usually, authorization is in the context of authentication: once you have authenticated a user, they may be authorized different types of access or activity.
Bastion Host	A system that has been hardened to resist attack, and which is installed on a network in such a way that it is expected to potentially come under attack. Bastion hosts are often components of firewalls, or may be "outside" Web servers or public access systems. Generally, a bastion host is running some form of general purpose operating system (e.g., UNIX, VMS, WNT, etc.) rather than a ROM-based or firmware operating system.
Challenge/Response	An authentication technique whereby a server sends an unpredictable challenge to the user, who computes a response using some form of authentication token.
Chroot	A technique under UNIX whereby a process is permanently restricted to an isolated subset of the filesystem.
Cryptographic Checksum	A one-way function applied to a file to produce a unique "fingerprint" of the file for later reference. Checksum systems are a primary means of detecting filesystem tampering on UNIX.
Data Driven Attack	A form of attack in which the attack is xencoded in innocuous-seeming data which is executed by a xuser or other software to implement an attack. In the case of firewalls, a data driven attack is a concern since it may get through the firewall in data form and launch an attack against a system behind the firewall.

[6] Entnommen aus: [15]

Defense in Depth	The security approach whereby each system on the network is secured to the greatest possible degree. May be used in conjunction with firewalls.
DNS spoofing	Assuming the DNS name of another system by either corrupting the name service cache of a victim system, or by compromising a domain name server for a valid domain.
Dual Homed Gateway	A dual homed gateway is a system that has two or more network interfaces, each of which is connected to a different network. In firewall configurations, a dual homed gateway usually acts to block or filter some or all of the traffic trying to pass between the networks.
Encrypting Router	see Tunneling Router and Virtual Network Perimeter.
Firewall	A system or combination of systems that enforces a boundary between two or more networks.
Host-based Security	The technique of securing an individual system from attack. Host based security is operating system and version dependent.
Insider Attack	An attack originating from inside a protected network.
Intrusion Detection	Detection of break-ins or break-in attempts either manually or via software expert systems that operate on logs or other information available on the network.
IP Spoofing	An attack whereby a system attempts to illicitly impersonate another system by using its IP network address.
IP Splicing / Hijacking	An attack whereby an active, established, session is intercepted and co-opted by the attacker. IP Splicing attacks may occur after an authentication has been made, permitting the attacker to assume the role of an already authorized user. Primary protections against IP Splicing rely on encryption at the session or network layer.
Least Privilege	Designing operational aspects of a system to operate with a minimum amount of system privilege. This reduces the authorization level at which various actions are performed and decreases the chance that a process or user with high privileges may be caused to perform unauthorized activity resulting in a security breach.
Logging	The process of storing information about events that occurred on the firewall or network.
Log Retention	How long audit logs are retained and maintained.
Log Processing	How audit logs are processed, searched for key events, or summarized.
Network-Level Firewall	A firewall in which traffic is examined at the network protocol packet level.
Perimeter-based Security	The technique of securing a network by controlling access to all entry and exit points of the network.

Policy	Organization-level rules governing acceptable use of computing resources, security practices, and operational procedures.
Proxy	A software agent that acts on behalf of a user. Typical proxies accept a connection from a user, make a decision as to whether or not the user or client IP address is permitted to use the proxy, perhaps does additional authentication, and then completes a connection on behalf of the user to a remote destination.
Screened Host	A host on a network behind a screening router. The degree to which a screened host may be accessed depends on the screening rules in the router.
Screened Subnet	A subnet behind a screening router. The degree to which the subnet may be accessed depends on the screening rules in the router.
Screening Router	A router configured to permit or deny traffic based on a set of permission rules installed by the administrator.
Session Stealing	See IP Splicing.
Trojan Horse	A software entity that appears to do something normal but which, in fact, contains a trapdoor or attack program.
Tunneling Router	A router or system capable of routing traffic by encrypting it and encapsulating it for transmission across an untrusted network, for eventual deencapsulation and decryption.
Social Engineering	An attack based on deceiving users or administrators at the target site. Social engineering attacks are typically carried out by telephoning users or operators and pretending to be an authorized user, to attempt to gain illicit access to systems.
Virtual Network Perimeter	A network that appears to be a single protected network behind firewalls, which actually encompasses encrypted virtual links over untrusted networks.
Virus	A self-replicating code segment. Viruses may or may not contain attack programs or trapdoors.

8.5 Literatur

[1] S.Bellovin — Security Problems in the TCP/IP Protocol Suite, AT&T Bell Laboratories, Computer communications Review Vol.19, No.2, pp32–48, April 1989

[2] TIS Firewall Toolkit, Teil der Dokumentation, http://www.tis.com, Jan. 1995

[3] TIS Firewall Toolkit Overview, Teil der Dokumentation, http://www.tis.com, Jan. 1995

[4] M.Ranum, F.Avolio — A Toolkit and Methods for Internet Firewalls, Teil der Dokumentation,
http://www.tis.com, Jan. 1995

[5] R.Niederberger — Firewalls: Sicherheit und Benutzerakzeptanz in Forschungsnetzen, Interner Bericht, Forschungszentrum Jülich, KFA-ZAM-IB-9522, Verfügbar mittels E-Mail an: dispatch.zam@KFA-Juelich.de

[6] D.B.Chapman, E.D.Zwicky — Building Internet Firewalls, O'Reilly & Associates, 1st Edition, ISBN 1–56592–124–0, Sept. 1995

[7] Cert Advisory — IP Spoofing Attacks and Hijacked Terminal Connections, CA-95:01, Jan. 23 1995

[8] Ph.Karn, N.M.Haller, J.S.Walden, S.Chasin — S/Key Authentication, ftp://thumper.bellcore.com, in /pub/nmh/skey

[9] Security Dynamics – Securing the Information Age ... Minute by Minute, Security Dynamics Technologies, 101A4 20M 4/1/95

[10] D.Vinzencetti — STEL: Secure TELnet, Fifth USENIX Unix Security Symposium, Salt Lake City,UT, June 1995

[11] D.I.Dalva — Security and the World Wide Web, Trusted Information Systems, Inc., http://www.tis.com, Aug. 12 1994

[12] B.Cheswick, S.Bellovin — Firewalls and Internet Security : Repelling the Wily Hacker, Addison-Wesley Professional Computer Series, ISBN 0–201–63357–4, Jul. 1994

[13] B.Cheswick — The Design of a Secure Internet Gateway, USENIX proceedings. Verfügbar mittels FTP: ftp://research.att.com, /dist/secure_internet_gateway.ps

[14] M.Ranum — Thinking About Firewalls, Proceeding of the Second World Conference on Systems Management and Security (SANS II), 1993, Verfügbar mittels FTP: ftp://ftp.tis.com, /pub/firewalls/firewall.ps.Z

[15] M.Ranum — Firewall Product Functional Summary, Information Warehouse Inc., ftp://ftp.iwi.com, Aug. 1995